|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **POLITEKNIK NEGERI MALANG**  **JURUSAN AKUNTANSI**  **PROGRAM STUDI : D3 MANAJEMEN INFORMATIKA** | | | | | | | | | | | | |
| **RENCANA PEMBELAJARAN SEMESTER (RPS)** | | | | | | | | | | | | | | | |
| **MATA KULIAH** | | | **KODE** | **RUMPUN MATA KULIAH** | | | | | **BOBOT (sks)/jam** | | **SEMESTER** | | **TGL. PENYUSUNAN** | | |
| **PEMROGRAMAN BERORIENTASI OBJEK** | | | **RIF 4** | **INFORMATIKA DASAR** | | | | | **2 sks / 3 jam** | | **2** | | **21 Februari 2017** | | |
| **OTORISASI** | | | **Dosen Pengembang RPS** | | | | | | **Koordinator RMK** | | **Ka PRODI** | | | | |
| **Imam Fahrur Rozi, ST., MT.** | | | | | | **Mungki Astiningrum, ST., M.Kom.** | | **Dr. Eng. Rosa Andrie A., ST., MT.** | | | | |
| **Capaian Pembelajaran (CP)** | | | **Capaian Pembelajaran Lulusan Program Studi (CPL-Prodi)** | | | | | | | |  | | | | |
| **…………………………………………………**  **…………………………………………………**  **…………………………………………………**  **…………………………………………………**  **…………………………………………………** | | | | | | | | | | | | |
| **Capaian Pembelajaran Lulusan yang dibebankan pada mata kuliah (CPL-MK)** | | | | | | | | | | |  | |
| **………………………………………………….**  **………………………………………………….**  **………………………………………………….** | | | | | | | | | | | | |
| **Diskripsi Singkat Mata Kuliah** | | | Mata kuliah Pemrograman Berbasis Objek (PBO) adalah matakuliah yang mengajarkan kepada mahasiswa bagaimana konsep pengembangan program dengan paradigma berorientasi objek | | | | | | | | | | | | |
| **Materi Pembelajaran / Pokok Bahasan** | | | * Konsep OOP * Class * Object * Enkpasulasi * Inheritance * Polimorfisme * Abstract Class * Interface * Java Basic Programming * Pengenalan GUI * Pengenalan Java API | | | | | | | | | | | | |
| **Pustaka** | | | **Utama :** | |  | | | | | | | | | | |
| Horstmann, C. S., & Cornell, G. (2007). *Core Java Volume I–Fundamentals, Eighth Edition.* Network Circle, Santa Clara: Prentice Hall.  Horstmann, C. S., & Cornell, G. (2008). *Core Java Volume II–Advanced Features, Eighth Edition.* Network Circle, Santa Clara: Prentice Hall. | | | | | | | | | | | | |
| **Pendukung :** | |  | | | | | | | | | | |
| Rickyanto, I. (2005). *Dasar Pemrograman Berorientasi Objek dengan Java 2.* Yogyakarta: Andi Offset | | | | | | | | | | | | |
| **Media Pembelajaran** | | | **Software :** | | | | **Hardware :** | | | | | | | | |
| JDK 1.8  Netbeans 8 | | | | Komputer/Laptop dengan minilmal RAM 512 MB | | | | | | | | |
| **Nama Dosen Pengampu** | | | Imam Fahrur Rozi, ST., MT. | | | | | | | | | | | | |
| **Matakuliah Syarat** | | | Algoritma dan Pemrograman | | | | | | | | | | | | |
| **Minggu Ke** | **Kemampuan Akhir Yang Direncanakan**  **(Sub-CP-MK)** | **Bahan kajian**  **(Materi Pembelajaran)** | | | | **Bentuk dan Metode Pembelajaran** | | **Estimasi Waktu** | | **Pengalaman Belajar Mahasiswa** | | **Kriteria & Bentuk Penilaian** | | **Indikator Penilaian** | **Bobot (%)** |
| **(1)** | **(2)** | **(3)** | | | | **(4)** | | **(5)** | | **(6)** | | **(7)** | | **(8)** | **(9)** |
| 1 | Pengantar | 1. Pembahasan silabus dan kontrak kuliah 2. Pengenalan Greenfoot 3. Simulasi pembuatan skenario permaian sederhana di Greenfoot | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi dari dosen dan berdiskusi * Menyebutkan komponen/objek yang terlibat dalam suatu sistem * Menyebutkan objek yang terlibat dalam skenario game sederhana yang akan disimulasikan menggunakan Greenfoot | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes Lisan | | * Mampu menyebutkan dengan tepat objek world dan objek aktor yang terlibat dalam skenario game sederhana |  |
| 2 | Konsep Dasar OOP | 1. Pengenalan PBO 2. Perbedaan paradigma berorientasi objek dengan paradigma struktural 3. Konsep dasar PBO 4. Class 5. Object 6. Enkapsulasi 7. Inheritance 8. Polimorfisme 9. Pengenalan pemodelan UML | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi dari dosen dan berdiskusi * Menyebutkan komponen-komponen sistem (misal: SIakad) baik dari sudut pandang OOP maupun struktural * Menyebutkan objek dan ciri-ciri tiap objek (apa yang bisa dilakukan dan apa yang dimiliki oleh objek) yang ada di suatu tempat/sistem * Menyebutkan objek-objek yang memiliki kesamaan sifat dan merupakan turunan dari objek lain * Menyebutkan objek yang bisa memiliki banyak bentuk | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes Lisan | | * Ketepatan dalam menjelaskan perbedaan antara paradigma OOP dan Struktural * Ketepatan dalam menjelaskan konsep Class, object, enkapsulasi, inheritance dan polimorfisme * Ketepatan dalam memberikan contoh dari masing-masing konsep dasar OOP |  |
| 3 | Class dan Object | 1. Class 2. Object 3. Atribut 4. Method 5. Instansiasi 6. Pengaksesan Atribut dan Method 7. UML:Diagram Class | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi dari dosen dan berdiskusi * Membentuk kelompok kecil dan berdiskusi dalam kelompok untuk mengidentifikasi objek (lengkap dengan atribut dan method) yang ada dalam suatu sistem yang ditentukan * Menggambarkan objek tersebut menggunakan diagram class * Mempresentasikan hasil diskusi dalam kelas | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes Tulis | | * Ketepatan dalam mengidentifikasi objek * Ketepatan dalam menentukan atribut dan method dari tiap objek yang telah diidentifikasi * Ketepatan dalam penggambarannya ke dalam diagram class * Kemampuan dalam presentasi | 2.5% |
| 4 | Dasar Pemrograman Java | 1. Tipe data 2. Variabel 3. Seleksi Kondisi 4. Perulangan 5. Array 6. pengenalan try-catch (ArrayIndexOutOfBoundException, ArithmaticException, NullPointerException) | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi dari dosen dan berdiskusi * Membentuk kelompok kecil dan menyelesaikan masalah sederhana dengan menggunakan konsep dasar pemrograman Java yang telah dijelaskan | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes Tulis | | * Ketepatan dalam membuat program untuk menyelesaikan suatu kasus, dengan menggunakan konsep dasar pemrograman Java yang telah dijelaskan |  |
| **5** | **Kuis 1** | Materi pertemuan 1-4 | | | | **Ujian Tulis/Online** | | **2x45’** | | * Menjawab pertanyaan/soal Kuis | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes Tulis/Online | | * Ketepatan dalam menjawab pertanyaan | **15%** |
| 6 | Enkapsulasi | 1. Konstruktor 2. Access modifier 3. Atribut/Method Class 4. Atribut/Method Instansiasi 5. Setter dan getter 6. UML:Notasi access modifier dan notasi static | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Mensimulasikan cara kerja konstruktor * Menentukan modifier yang tepat untuk atribut dan method dari suatu class yang telah dibuat/diberikan sebelumnya * Berdiskusi tentang perbedaan antara atribut instansiasi dan atribut class, antara method instansiasi dan method class * Berdiskusi tentang kegunaan setter dan getter | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes lisan, tulis | | * Ketepatan dalam menentukan konstruktor dan ketepatan dalam melakukan instansiasi yang menggunakan konstruktor tersebut * Ketepatan dalam menentukan modifier access dari suatu atribut dan method * Ketepatan dalam menjelaskan perbedaan atribut/method instansiasi dan atribut/method class * Ketepatan dalam membuat setter dan getter * Ketepatan dalam penulisan notasi static, dan modifier access |  |
| 7 | Relasi Class | 1. Relasi Has-A (Studi kasus 1 class memiliki hubungan has-a dengan 1 objek dari klass lain) 2. Penggambaran relasi class dengan diagram class | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Menuliskan program dari diagram class yang telah diberikan * Menggambarkan desain diagram class dari kasus yang diberikan, dengan berdiskusi dalam kelompok * Presentasi hasil perancangan di kelas | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes lisan, tulis | | * Ketepatan dalam menerjemahkan diagram class ke dalam program * Ketepatan dalam merancang diagram class dari suatu kasus tertentu * Kemampuan dalam presentasi | 2.5% |
| 8 | Relasi Class | 1. Relasi Has-A (Studi kasus 1 class memiliki hubungan has-a dengan lebih dari 1 objek dari klass lain) 2. Penggambaran relasi class dengan diagram class | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Menuliskan program dari diagram class yang telah diberikan * Menggambarkan desain diagram class dari kasus yang diberikan, dengan berdiskusi dalam kelompok * Presentasi hasil perancangan di kelas | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes lisan, tulis | | * Ketepatan dalam menerjemahkan diagram class ke dalam program * Ketepatan dalam merancang diagram class dari suatu kasus tertentu |  |
| **9** | **UTS** | Materi pertemuan 6-8 | | | | **Ujian Tulis/Online** | | **2x45’** | | **Menjawab pertanyaan/soal** | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes tulis/online | | * Ketepatan dalam menjawab pertanyaan | **30%** |
| 10 | Inheritance | 1. Pengertian Inheritance 2. Single dan Multilevel Inheritance 3. Super keyword 4. UML:relasi inheritance | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Diskusi dalam kelompok untuk membuat contoh class-class yang memiliki relasi inheritance (single dan multilevel), dan menggambarkannya dalam diagram class * Mempresentasikan hasil diskusi di kelas | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes lisan, tulis | | * Ketepatan dalam mengidentifikasi class yang memiliki relasi inheritance * Ketepatan membuat desain class yang memiliki relasi inheritance * Kemampuan presentasi |  |
| 11 | Overriding dan Overloading | 1. Overriding 2. Overloading | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Diskusi dalam kelompok untuk membuat contoh overriding dan overloading method * Mempresentasikan hasil diskusi di kelas | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes lisan, tulis | | * Ketepatan dalam mengidentifikasi method overriding dan overloading * Kemampuan presentasi | 2.5% |
| 12 | Abstract Class dan Interface | 1. Konsep Abstract Class 2. Abstract method 3. UML:notasi abstract 4. Konsep Interface 5. Beda Interface dan Abstract Class 6. Pembuatan interface 7. Cara membuat class yang implements ke interface 8. UML:notasi interface dan relasi implements | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Diskusi dalam kelompok untuk membuat contoh overriding dan overloading method * Mempresentasikan hasil diskusi di kelas | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes lisan, tulis | | * Ketepatan dalam merancang interface dan abstract class * Ketepatan dalam pemodelannya dalam diagram class * Kemampuan presentasi | 2.5% |
| 13 | Polimorfisme | 1. Konsep polimorfisme 2. heterogeneous collection 3. polymorphic arguments 4. virtual method invocation 5. casting object | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Memperhatikan simulasi | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes lisan | | * Ketepatan dalam menjelaskan penggunaan polimorfisme |  |
| 14 | **Kuis 2** | Materi pertemuan 10-13 | | | | **Ujian Tulis/Online** | | **2x45’** | | **Menjawab pertanyaan/soal** | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes tulis/online | | * Ketepatan dalam menjawab pertanyaan | **15%** |
| **15** | GUI | 1. Frame, Menu, Textfield, Button, Label, Combobox, Radiobutton, Checkbox 2. Event Handling (actionperformed) | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Memperhatikan simulasi * Merancang suatu tampilan GUI * Membuat Window * Menambahkan komponen-komponen GUI lainnya ke dalam window | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes tulis | | * Ketepatan dalam membuat window * Ketepatan dalam membuat komponen-komponen yang akan ditambahkan ke dalamnya |  |
| 16 | Java API | 1. Java Docs 2. String dan Collection (List, Set, Map) | | | | 1. Ceramah 2. Diskusi 3. Simulasi | | 3x45’ | | * Mendengarkan ceramah materi * Memperhatikan simulasi | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes lisan | | * Ketepatan dalam menjelaskan penggunaan dari Java Docs * Ketapatan dalam menjelaskan penggunaan dari String dan Collection |  |
| 17 | Tugas Besar | 1. Perancangan diagram class dari kasus yang diberikan | | | | Diskusi | | 3x45’ | | * Mengidentifikasi class dan menggambarkannya dalam diagram class * Berkonsultasi dengan dosen tentang hasil perancangannya | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes tulis | | * Ketepatan dalam mengidentifikasi class * Ketepatan dalam penggambaran diagram class |  |
| **18** | **UAS** | **Materi mulai pertemuan 1-16** | | | | **Ujian Online** | | **1x60’** | | **Menjawab pertanyaan/soal** | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes tulis/online | | * Ketepatan dalam menjawab pertanyaan | **30%** |
| **19** | **Remidi** | **Materi mulai pertemuan 1-16** | | | | **Ujian Tulis** | | 3x45’ | | **Menjawab pertanyaan/soal** | | Kriteria: Ketepatan dalam menjawab pertanyaan  Bentuk: tes tulis/online | | * Ketepatan dalam menjawab pertanyaan |  |

**Keterangan :**

-